

MENU

SEARCH

INDEX

DETAIL

E5388

1/1



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 11341076

(43) Date of publication of application: 10.12.1999

(51)Int.CL

H04L 12/66
G06F 17/30
H04L 12/56

(21)Application number: 10317235

(71)Applicant:

HITACHI LTD

(22)Date of filing: 09.11.1998

(72)Inventor:

SUKAI KAZUO

AIMOTO TAKESHI

MATSUYAMA NOBUHITO

AKAHA SHINICHI

SAKO YOSHITO

TANABE NOBORU

(30)Priority

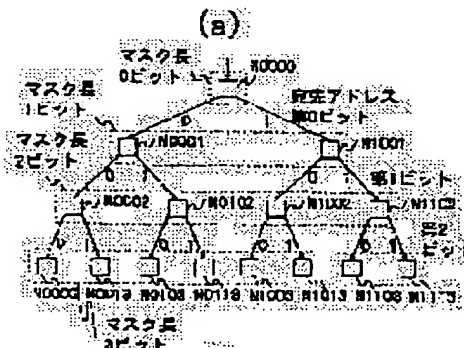
Priority number: 11999999 Priority date: 23.03.1998 Priority country: JP

(54) NETWORK REPEATER AND NEXT NETWORK TRANSFER DESTINATION RETRIEVAL METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To eliminate a time for retrieval processing by a host number bit of a sub-network address by extending a node by the host number bit of the sub-network address at the time of retrieval of a host bit correspondence part of the sub-network address.

SOLUTION: When nodes N0002, N0102, N1002 and N1102 of mask length 2 bit are extended at a fixed



(b)

図10(b)は、マスク長10ビットとアドレス10ビットの組み合わせを示しています。マスク長10ビットは、アドレス10ビットの上位10ビットと一致し、結果10ビットの上位10ビットを決定します。アドレス10ビットは、結果10ビットの下部10ビットを決定します。

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

[MENU](#)
[SEARCH](#)
[INDEX](#)
[DETAIL](#)

E5388

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-341076

(43) 公開日 平成11年(1999)12月10日

(51) Int.Cl.⁶

識別記号

F I

H 0 4 L 12/66

H 0 4 L 11/20

B

G 0 6 F 17/30

G 0 6 F 15/411

H 0 4 L 12/56

H 0 4 L 11/20

1 0 2 D

審査請求 未請求 請求項の数 11 O L (全 25 頁)

(21) 出願番号 特願平10-317235

(22) 出願日 平成10年(1998)11月9日

(31) 優先権主張番号 特願平11-999999

(32) 優先日 平10(1998)3月23日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 須貝 和雄

神奈川県秦野市堀山下1番地 株式会社日

立製作所汎用コンピュータ事業部内

(72) 発明者 相本 毅

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72) 発明者 松山 信仁

神奈川県秦野市堀山下1番地 株式会社日

立インフォメーションテクノロジー内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

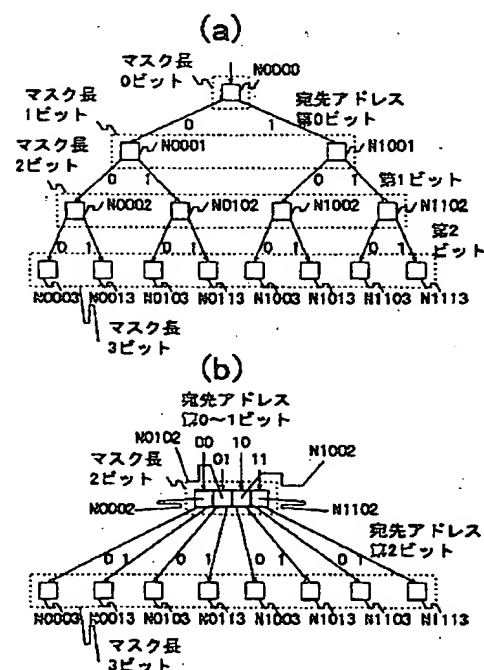
(54) 【発明の名称】 ネットワーク中継装置及びネットワーク次転送先検索方法

(57) 【要約】

【課題】 ルータ等のネットワーク中継装置において、パケットを転送するための転送際の経路検索処理を高速に実行する。

【解決手段】 経路情報に基づいてネットワーク中継装置に入ってきたパケットの宛先アドレスからパケットの転送先アドレスを検索する際に、経路検索のためのデータ構造を、受け取ったパケットの宛先アドレスの上位ビットから1ビットずつ検査してゆく2分木検索のp (pは2以上の整数) 段分を一つの2のp乗分木にし、2分木のp数段の検索を1段で行うにより、高速に経路を検索する。

図9



【特許請求の範囲】

【請求項1】複数のネットワークを接続するネットワーク中継装置であって、前記ネットワークの一つを接続するポートと、前記ポートに接続され、該ポートに接続されたネットワークとのインタフェースを制御するネットワークインタフェース部と、前記ネットワークインタフェース部と装置内通信路を介して接続され、前記ネットワークインタフェースから受け取ったパケットのルーティング処理を行うルーティング処理部とを有し、前記ルーティング処理部は、経路情報保持手段と、前記経路情報保持手段に保持された経路情報に基づいて前記受け取ったパケットの次に転送すべき経路を算出する次経路検索手段とを包含し、

前記次経路検索手段は、前記受け取ったパケットの宛先アドレスの上位ビットから1ビットずつ検査してゆく2分木検索の p (p は2以上の整数) 段分を、一つの2の p 乗分木にし、2分木の p 数段の検索を1段で行うことを特徴とするネットワーク中継装置。

【請求項2】前記次経路検索手段は、一つの2分木ノードと、その直下につながる $p-1$ 段分の合計2の p 乗-1個分の2分木ノードを一つの前記2の p 乗分木ノードにまとめ、まとめられる最下段の2の $(p-1)$ 乗個の2分木ノードに、それより上段のノードに割り付けられていた経路データを埋め込み、前記2の p 乗分木ノードを2分木を2の $(p-1)$ 乗個分併せた形で構成することを特徴とする請求項1記載のネットワーク中継装置。

【請求項3】前記次経路検索手段は、2分木を複数個併せるときに、一つだけ持てば良い要素を一つだけ持つようにすることを特徴とする請求項2記載のネットワーク中継装置。

【請求項4】前記次経路検索手段は、2の p 乗分木ノードを検索のために読むときにノード全てを読まずに、2の p 乗分木ノードを作成するときに併せた2の $(p-1)$ 乗個の2分木ノードの内の、いずれか一つに対応するデータのみを読むことを特徴とする請求項2記載のネットワーク中継装置。

【請求項5】前記次経路検索手段は、各ノードにそのノード自身のマスク長を格納せずにそのノードの直ぐ下に繋がるノードのマスク長を格納することにより、ノードのマスク長を、そのノードのデータを読む前に知り、宛先アドレスの、ノードのマスク長で示されるビット位置から、そのビット位置+ $p-1$ までの値に従い、ノードのデータの内の読み込む部分を選択することを特徴とする請求項2記載のネットワーク中継装置。

【請求項6】前記次経路検索手段は、各ノードの最初に読み込むデータ内に、そのノードに経路が割り付けられているか否かを示すフラグを設け、最初に、このフラグを読み込み、経路が割り付けられていないノードでは経路情報を読み込まないことを特徴とする請求項4記載のネットワーク中継装置。

【請求項7】前記ネットワーク中継装置はルータであることを特徴とする請求項1記載のネットワーク中継装置。

【請求項8】複数のネットワークを接続するネットワーク中継装置であって、前記ネットワークの一つを接続するポートと、前記ポートに接続され、該ポートに接続されたネットワークとのインタフェースを制御するネットワークインタフェース部と、前記ネットワークインタフェース部と装置内通信路を介して接続され、前記ネットワークインタフェースから受け取ったパケットのルーティング処理を行うルーティング処理部とを有し、前記ルーティング処理部は、経路情報保持手段と、前記経路情報保持手段に保持された経路情報に基づいて前記受け取ったパケットの次に転送すべき経路を算出する次経路検索手段とを包含し、

前記次経路検索手段は、次経路の検索を宛先アドレスの上位ビットから1ビットずつ検査してゆく2分木検索により行い、検査を行うビット位置を、マスク長に対応させることによりマスク付きの一致検索を行い、マスク長 m (m は自然数) ビットのノードを、2の m 乗個、記憶手段上の決まった位置に展開し、それぞれのマスク長 m ビットのノードを、それぞれ、宛先アドレスの第0ビットから第 $m-1$ ビットまでが取りうる値に1対1に対応させ、宛先アドレスの第0ビットから第 $m-1$ ビットの値に従い、マスク長 m ビットのノードの一つを選択することを特徴とするネットワーク中継装置。

【請求項9】複数のネットワークを接続するネットワーク中継装置であって、前記ネットワークの一つを接続するポートと、前記ポートに接続され、該ポートに接続されたネットワークとのインタフェースを制御するネットワークインタフェース部と、前記ネットワークインタフェース部と装置内通信路を介して接続され、前記ネットワークインタフェースから受け取ったパケットのルーティング処理を行うルーティング処理部とを有し、前記ルーティング処理部は、経路情報保持手段と、前記経路情報保持手段に保持された経路情報に基づいて前記受け取ったパケットの次に転送すべき経路を算出する次経路検索手段とを包含し、

前記次経路検索手段は、次経路の検索を宛先アドレスの上位ビットから1ビットずつ検査してゆく2分木検索により行い、マスク長0ビットから k ビットまでの2分木ノードを、先頭からのビット数が所定数の部分を前記次経路検索手段の内蔵記憶手段内に置き、マスク長 $k+1$ ビット以降の2分木ノードを検索手段の外部記憶手段内に置き、第0から第 k ビットまでの検索処理と、第 $k+1$ ビット以降の検索処理をパイプライン処理することを特徴とするネットワーク中継装置。

【請求項10】複数のネットワークを接続し、前記ネットワークの一つから受け取ったパケットを経路情報に基づいて次の転送先に送出するネットワーク中継装置にお

けるネットワーク次転送先検索方法であって、前記受け取ったパケットの宛先アドレスの上位ビットから1ビットずつ検査してゆく2分木検索の p (p は2以上の整数)段分を、一つの2の p 乗分木にし、2分木の p 数段の検索を1段で行うことを特徴とするネットワーク次転送先検索方法。

【請求項11】前記ルーティング処理部内次経路検索手段で行う次経路検索処理をハードウェアで行うことにより、ノード内のデータを読み出すアドレスの計算処理と経路情報の読み込み処理を並列に行うことを特徴とする請求項1乃至請求項9のいずれかに記載のネットワーク中継装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータネットワークシステムにおけるルータ等のネットワーク中継装置に関し、特にネットワーク中継装置に入ってきたパケットの宛先アドレスからパケットの転送先アドレスを高速に検索するのに適したネットワーク中継装置及びネットワーク次転送先検索方法に関する。

【0002】

【従来の技術】ネットワークシステムにおいては、複数のサブネットを接続するためにルータ等のネットワーク中継装置が用いられる。ルータは接続されているサブネットから受け取ったパケットの宛先アドレスを調べてパケットの転送先を決定し、転送先のルータやホストが接続されたサブネットに受け取ったパケットを転送する。図1は複数のサブネットがルータによって接続された一般的なネットワークシステムの構成を示す。

【0003】図1において、R1及びR2はルータ、SN1はルータR1のポートP11に接続されたサブネットワーク、SN2はルータR1のポートP12及びルータR2のポートP21に接続されたサブネットワーク、H10及びH11はサブネットSN1に接続されたホスト、H20及びH21はサブネットSN2に接続されたホスト、H30及びH31はサブネットSN3に接続されたホストである。

【0004】ホストH10からホストH21にパケットを送る場合、ルータR1はパケット内にヘッダ情報として格納されている宛先アドレスDAを調べて、宛先のホストH21がサブネットSN2上にあり、かつサブネットSN2がルータR1に直接接続されていることを認識する。そして、ルータR1はパケットをサブネットSN2に接続されているポートP12に出力し、出力するときに次に転送するアドレス(次ホップアドレス)を宛先ホスト自身(H21)とする。

【0005】また、ホストH10からホストH31にパケットを送る場合、ルータR1はパケット内にヘッダ情報として格納されている宛先アドレスDAを見て、宛先のホストH31がサブネットSN3上にあり、かつサブ

ネットSN3がルータR1に直接には接続されておらず、ルータR2経由で接続されていることを認識する。ルータR1はパケットをルータR2が接続されているサブネットSN2に接続されているポートP21に出力し、出力するときに次に転送するアドレス(次ホップアドレス)をルータR2とする。この場合、ルータR2はパケットを受け取ると、ルータR1と同様に、宛先アドレスDAを見てパケットをホストH31に転送する。

【0006】次に、ルータがパケットを受け取ったときに、次に転送するアドレス、及びパケットを出力するポートを検索するときの検索仕様を図2を使用し説明する。TBLは経路検索テーブルであり、このテーブルは人手で入力された構成定義情報、及びルータ間での接続情報のやりとりにより得られた情報から作成される。

【0007】経路検索テーブルTBLは、サブネットワークアドレスとサブネットワークマスク長の組を検索のキーとして、出力ポート、次ホップアドレス、及びサブネットワークが直接接続されているか否かの情報(以後、次ホップ情報と呼ぶ)を検索するものである。

【0008】経路検索仕様においては、最上位ビット側からサブネットワークマスク長のビット数だけを有効とするマスクを宛先アドレスに掛けたものをサブネットワークアドレスと比較する。比較の結果、一般的にはマスク長の異なる複数のエントリE1、E2、E4が一致し、一致したエントリの中でマスク長が最長のものE2の次ホップ情報(次ホップ2)を検索結果とする。

【0009】この検索仕様に従った検索を高速に行う方法として、Radishアルゴリズムがある。Radishアルゴリズムについては例えば、UNIX MAGAZINE 1997. 4 pp. 20-25 山口英「カーネルを読もう(8)IP層における経路制御機構(2)」に解説されている。

【0010】

【発明が解決しようとする課題】上記Radishアルゴリズムは、左右にポイントを持つ複数の頂点(ノード)をポイントでつないだ木から構成される木構造の各ノードに経路エントリをマップし、この木を辿るときには、各ノードの左右のどちらかのポイントを辿り次のノードに移動することにより、目的の経路エントリがマップされたノードにたどり着くアルゴリズムである。

【0011】まず、図3を参照して木の構造を説明する。考え方はビット長には依存しないので、図3では理解し易いようアドレス長を3ビットとして説明する。

【0012】図3に示すように、各ノードを、木の上から順にマスク長0ビット、1ビット、2ビット、3ビットのノードと呼ぶ。

【0013】マスク長0ビットのノードN0000では宛先アドレスの第0ビットが0か1かに従い左/右のポイントを辿ることによりマスク長1ビットのノードN0001、N1001に移り、マスク長1ビットのノード

【0023】ここで、図8に示す例では、第0ビットは

既にテストされ、第0ビットが等しくなる方の分岐が選択されている為、常に一致する。一般に、あるノードにたどり着く毎に正しいノードにたどり着いたか否かを検査していれば、第0ビットからそのノードのマスク長までのビットは宛先アドレスとノードのサブネットワークアドレスとで等しいことが保証されているので、次のノードにたどり着いたときに、前にどのビットまでテストしたかに関らず、第0ビットからノードのマスク長までのビットが宛先アドレスとノードのサブネットワークアドレスとで等しいか否かを調べて良い。

【0024】このように、Radishアルゴリズムでは、経路を検索するために宛先アドレスを上から1ビットずつ検査しており、経路検索処理に時間が掛かる、という問題があった。

【0025】本発明の目的は、パケットを転送するための転送際の経路検索処理を高速に実行するネットワーク中継装置、特にルータを提供することにある。

【0026】本発明の他の目的は、ルータ等のネットワーク中継装置において、受信したパケットの宛先アドレスからパケットの転送先アドレスを高速に検索するネットワーク次転送先検索方法を提供することにある。

【0027】

【課題を解決するための手段】上記目的を達成するため、本発明においては、サブネットワークアドレスの上位ビットに対応する部分の検索について、サブネットワークアドレスの上位数ビット分、ノードをメモリ上の決まった位置に展開することにより、サブネットワークアドレスの上位数ビット分の検索処理時間を無くすようにする。

【0028】また、検索木の、上位数ビット分を検索処理を行うLSIに内蔵し、LSI内部のメモリと外部のメモリとの間で検索処理をパイプライン処理することにより、上位数ビット分の検索処理時間を隠すようにする。また、検索木を構成する各ノードを、従来技術での2分木ノードから4分木、8分木、あるいはそれ以上と枝別れの数 2 のべき乗で増やし、一つのノードで1ビットでなく、連続する2ビット、3ビット、あるいはそれ以上のビット数を同時に検査し、検索終了までに辿るノードの数を減らすようにする。

【0029】また、検索木を表現するデータ構造を記憶するためのメモリ量を減らすために、4分木、8分木、あるいは一般に2の p 乗分木を構成するとき、一つの2分木ノードと、その直下につながる $p-1$ 段分の合計2の p 乗 -1 個分の2分木ノードを一つの2の p 乗分木ノードにまとめ、まとめられる最下段の2の $(p-1)$ 乗個の2分木ノードに、それより上段のノードに割り付けられていた経路データを埋め込むことにより、2の p 乗分木ノードを2分木を2の $(p-1)$ 乗個分併せた形で構成するようにし、さらに、2分木を複数個併せるときに、一つだけ持てば良い要素の一つだけ持つようにす

る。

【0030】また、この2分木を複数個併せた形で構成した4分木、8分木、あるいはそれ以上の枝別れ数のノードを、検索のために読むときにノード全てを読むのではなく、必要な部分のみを読むようにし、ノードが大きくなることによるデータの読み込み時間の増大を防ぐ。ノードのデータの内、必要な部分のみを選択するため、ノードのマスク長をノードのデータを読む前に知る必要があり、各ノードには、そのノードのすぐ下につながるノードのマスク長を格納するようにする。また、各ノードの先頭に、そのノードに経路が割り付けられているか否かを示すフラグを設け、最初に、このフラグを読み込み、経路が割り付けられていないノードでは、経路情報を読み込まないようにすることにより、データの読み込み時間の短縮を図る。

【0031】また、木構造の検索処理を専用ハードウェアで行うことにより、次のノードの読み込む位置のアドレス計算と、経路情報の候補の更新処理を、並列化することにより高速化する。

【0032】

【発明の実施の形態】本発明をより詳細に説明するため、添付の図面を参照して本発明を実施するための最良の形態を説明する。

【0033】最初に、本発明が適用されるルータ装置の代表的な構成を図38を参照して説明する。図38において、100はルータ装置、110はルーティング制御部、120はルータバス、130はネットワークインタフェース部、140はポート、150はサブネットワークである。

【0034】ネットワークインタフェース部130は、ポート140に接続されたサブネットワークからパケットを受け取り、受け取ったパケットをルータバス120経由でルーティング制御部110に送信する。ルーティング制御部110はルーティング情報を保持するルーティングテーブルを備え、このルーティング情報を用いて受け取ったパケットの宛先から転送先のサブネットワーク150を決定し、当該サブネットワーク150が接続されるポート140のネットワークインタフェース部130にパケットを送信する。ルーティング制御部110からパケットを受け取ったネットワークインタフェース部130はそのパケットを転送先のサブネットワーク150に送出する。なお、ルーティング制御部110は、受け取ったパケットのヘッダ情報に基づいてルーティングテーブルに保持するルーティング情報を更新・保守するとともに、ルータ装置100全体の管理機能を備えている。

【0035】図39は、ルータ装置の他の構成例を示すブロック図である。図39において、200はルータ装置、210はルーティングプロセッサ(RP)、220はルータ装置内通信手段、230はネットワークインタ

フェース部、240はポート、250はサブネットワーク、260はルータ管理部である。本構成の場合、図38に示した構成のルーティング制御部110がルーティング機能を実行するルーティングプロセッサ210及びルータ装置200の管理を行うルーティング管理部260に分かれるとともに、図38に示した構成に相当するネットワークインタフェース部230及びルーティングプロセッサ210から構成される部分を複数備えている。ルーティング管理部260は、ルータ装置200全体の管理機能を備えるとともに、各ルーティングプロセッサ210にルーティング情報を配付する。ルータ装置内通信手段220は、クロスバスイッチあるいはバス等であり、ルーティングプロセッサ210相互の通信やルーティングプロセッサ210とルーティング管理部260との間の通信を行う。ルーティングプロセッサ210は、図38のルーティング制御部110と同様に自分に接続されたネットワークインタフェース部230の間のパケット転送を行うとともに、他のルーティングプロセッサ210に接続されたサブネットワーク250にパケットを転送する場合は、ルータ装置内通信手段220を介して該当するルーティングプロセッサ210にパケットを転送する。

【0036】次に、ルーティング制御部110及びルーティングプロセッサ210において実行される次転送先経路検索処理について説明する。最初に、Radishアルゴリズムの本発明による高速化法を示す高速化の1番目の方法を図9を参照して説明する。従来のRadishアルゴリズムでは、木が縮退していない場合、マスク長0ビットのノードから順に1ビットずつ検索してゆくが、本発明では図9に示すように、マスク長 m ビットのノードをノードが有る場合も無い場合も、全てメモリ上の決まった位置に展開するものである。図9はマスク長2ビットのノードN0002、N0102、N1002、N1102をメモリの決まった位置に展開した場合の例である。

【0037】この場合、従来はまずマスク長0ビットのノードN0000に跳び、第0ビットの値に従い、マスク長1ビットのノードN0001、N1001のどちらかに跳び、第1ビットの値に従い、マスク長2ビットのノードN0002、N0102のどちらか、あるいはN1002、N1102のどちらかにたどり着いていた。

【0038】本発明においては、第0ビット及び第1ビットの値からマスク長2ビットのノードN0002、N0102、N1002、N1102を展開してあるアドレスを求め、直接マスク長2ビットのノードN0002、N0102、N1002、N1102のいずれかに跳ぶ。これにより、2回のノード検索の時間分、検索時間が短縮される。

【0039】一般に、マスク長 m ビットのノードをメモリ上に決まった位置に展開し、1回で跳んだ場合、マス

ク長0ビットから $m-1$ ビットまでのノードの計 m 回のノードを渡る時間分、検索時間が短縮される。一方で、2の m 乗個のマスク長 m ビットのノードをノードが有る場合も無い場合もメモリ上に展開する必要があるため、メモリ効率が悪くなる。したがって、メモリ効率と性能のトレードオフから m の値を決めるようにする

次に高速化の2番目の方法を図10乃至図12を参照して説明する。図10に示すようにマスク長 k ビットまでのノードを経路検索を行うLSI L1の内蔵メモリML1に展開し、マスク長 $k+1$ ビット以降のノードをLSI L1外部のメモリM1に展開する。このようにすることにより、LSI L1内部のメモリML1のアクセスが高速であること、及びLSI L1内部のメモリML1と外部のメモリM1とが独立したメモリであることから、パイプライン処理できることを利用し、高速化を図るものである。一般的に、LSI L1に内蔵できるメモリ量は外部に持つことができるメモリ量に比べ少ないが、マスク長が短い内は、ノードの数が少ないという性質があるので、マスク長が短い方のノードをLSI L1に内蔵することができる。

【0040】図11にパイプライン処理を行っていない従来の場合のタイムチャートを、図12に本願発明によるLSI L1内部のメモリML1と外部のメモリM1との間で経路検索処理のパイプライン処理を行っているときのタイムチャートを各々示す。従来は図11に示すように、あるパケット（パケット1、2、3）の経路検索を行う場合に、パケット1のマスク長 k ビットまでのノードの検索処理PR10と、マスク長 $k+1$ ビット以降のノードの検索処理PR11、パケット2の同上の処理PR20、PR21、パケット3の同上の処理PR30、PR31を順番に行っていた。これに対して、本発明では図12に示すように、LSI L1内部のメモリML1を使用してパケット1のマスク長 k ビットまでのノードの検索処理PR10を行った後、LSI L1外部のメモリM1を使用しパケット1のマスク長 $k+1$ ビット以降のノードの検索処理PR11を始めると同時に、LSI L1内部のメモリML1を使用しパケット2のマスク長 k ビットまでのノードの検索処理PR20を始めようとする。その後、処理PR20及びPR11が終わったら、LSI L1外部のメモリM1を使用しパケット2のマスク長 $k+1$ ビット以降のノードの検索処理PR21を始めると同時に、LSI L1内部のメモリML1を使用しパケット3のマスク長 k ビットまでのノードの検索処理PR20を始める。以後、同様にパケットの検索処理をパイプライン処理で行う。

【0041】高速化の3番目の方法は、従来一つのノードに2つの分岐先があり1ビットずつ検索していたものを、一つのノードに2の p 乗の分岐先を設け、同時に p ビットずつ検索することにより、従来に比べ検索時間を $1/p$ に短縮するものである。以後、一つのノードに2

の p 乗の分岐先があるノードのことを2の p 乗分木ノードと呼ぶ。

【0042】2の p 乗分木ノードは、従来の方法である2分木ノードから構成される木を変形することにより作成する。木の変形の方法は、一つの第 n ビット目の2分木ノードと、この2分木ノードの下第 $n+1$ から $n+p-1$ ビット目の2分木ノードを、一つの2の p 乗分木ノードに対応させるように行う。例として、2分木から8分木への変形法を図13乃至図15に示す。

【0043】8分木の場合、一つの8分木に対応させる2分木ノードのビット位置のとり方として、以下の3通りがある。

【0044】(a) 図13に示すように、マスク長0～2、3～5、6～8、9～11、12～14、15～17、18～20、21～23、24～26、27～29、30～32ビットのノードを、それぞれ一つの8分木ノードとする場合。

【0045】(b) 図14に示すように、第1～3、4～6、7～9、10～12、13～15、16～18、19～21、22～24、25～27、28～30、31～32ビットのノードを、それぞれ一つの8分木ノードとする場合。

【0046】(c) 図15に示すように、第2～4、5～7、8～10、11～13、14～16、17～19、20～22、23～25、26～28、29～31、32ビットのノードを、それぞれ一つの8分木ノードとする場合。

【0047】どの区切り方でも構成可能だが、木全体では、経路の追加、削除を容易に行えるように、上記3通りのビット位置の区切り方の内の一つを使用する。

【0048】上記3通りのビット位置の区切り方の内、最初の区切り方以外ではマスク長が0ビットから始まっていないので、最初のビットの検索を別に行う必要がある。この検索には、図9に示したマスク長 m ビットのノードをメモリ上に展開する方法を使用する。図14、図15に示した区切り位置に対応する8分木の最初のビット分のノードをメモリ上に展開した木の構成を図16、図17に示す。

【0049】図16に示す構成では、マスク長1～3ビットのノードN8013、N8113をメモリ上の決まった位置に並べ、それぞれを第0ビットが0か1かに従って選択する。図17に示す構成では、マスク長2～4ビットのノードN80024、N80124、N81024、N81124をメモリ上の決まった位置に並べ、それぞれを第0～1ビットが00か01か10か11かに従って選択する。

【0050】あるいは、図9に示したマスク長 m ビットのノードをメモリ上に展開する方法と、2の p 乗分木ノードにして複数のビット数を同時に検索する方法とを組み合わせることも可能である。具体的には、図13、図

14、図15に示す各ビットの区切り位置の場合に、それぞれ最初に並べるノード数を1、2、4個ではなく、これらの8倍である8、16、32個、8の2乗倍である64、128、256個、或いは一般に8の q 乗倍個にし、最初の1回、2回、或いは一般に2の p 乗分木ノードの q 回の検索時間を無くすることも可能である。 $p=3$ 、 $q=1$ の場合、即ち8分木ノードの1回のノードの検索時間を無くす場合で、図13、図14、図15に示す3通りの各ビットの区切り位置の場合についてのメモリ上へのノードの展開法を、図18、図19、図20にそれぞれ示す。このように p 、 q の値を大きくすると経路検索時間を短縮することができるが、多くのメモリを必要とするので、 p 、 q の値はメモリ効率と性能のトレードオフから決めるようにする。

【0051】次に、4分木ノード、8分木ノード、16分木ノード、或いは一般に2のべき乗分木のノードの構成法を図21を使用し説明する。

【0052】図21は4分木で、ある一つの2分木ノードA、B、C、D、Eと、その直下の2個の2分木ノードA0、A1、B0、B1、C0、C1、D0、D1、E0、E1の、各々合計3個の2分木ノードをまとめて一つの4分木ノードN401、N4023、N4123、N4223、N4323にする場合の例であり、合計3個の2分木ノードをつぶして、下の方の2分木ノードだけの大きさにする。つぶし方は、マスク長が異なる複数の経路が一致したら、マスク長の長いほうの経路を採用するという経路検索の仕様に従い、経路検索を行った場合に、2分木の場合と4分木の場合とで、経路検索結果が同じになる、という条件を満たすように行う。

【0053】4分木の場合の、このノードのつぶし方を図22～図30に示す。

【0054】3つのノードが全部ある場合(図22)、全ノードに経路情報が割り付けられていたら、下のノードの経路情報*A0、*A1を残し、上のノードの経路情報*Aは削除する。これは、ノードAの経路が一致したらノードA0かノードA1のどちらかの経路が必ず一致するので、マスク長が異なる複数の経路が一致したら、マスク長の長いほうの経路を採用するという経路検索の仕様により、*Aが使われることが無いからである。

【0055】上のノードAに経路情報*Aが割り付けられており、下のノードA0、A1の内A1にだけ経路が割り付けられていない場合(図23)、A1の経路情報に、Aの経路情報*Aを入れる。下のノードA0、A1の内A0だけ経路が割り付けられていない場合も同様である。

【0056】下のノードA0、A1の両方に経路が割り付けられていない場合(図24)には、A0、A1の両方の経路情報に、Aの経路情報*Aを入れる。

【0057】下のノードA1が無い場合(図25)は当

該ノードを補い、経路情報には上のノードAの経路情報 * Aを入れ、ノードA1の下にはノードが繋がっていないので、ノードA1の下ノードへのポインタにはNULLを入れる。下のノードA0、A1の内A0だけ経路が割り付けられていない場合も同様である。

【0058】下のノードA0、A1の両方が無い場合（図26）、両方を補い、両方の経路情報に、Aの経路情報 * Aを入れ、両方の下ノードへのポインタにはNULLを入れる。

【0059】上のノードに経路が割り付けられていない場合（図27）、上のノードを只つぶす。

【0060】上のノードAに経路情報 * Aが割り付けられてなく、下のノードA0、A1の内、A1にも経路が割り付けられていない場合（図28）、4分木にした場合もA1の経路情報は無い。下のノードA0、A1の内、A0に経路が割り付けられていない場合も同様である。

【0061】下のノードA0、A1の両方に経路が割り付けられていない場合（図29）、4分木にした場合も両方の経路情報は無い。

【0062】下のノードA0だけしかない場合（図30）には、下のノードA1を補う。下のノードA1だけしかない場合も同様である。

【0063】8分木の場合も同様にして、一つにまとめる7個の2分木ノードをつぶして、一番下の4個のノードだけの大きさにする。上の方の3つの2分木ノードのつぶし方の例を2つ図31に示す。

【0064】図31（a）は一つにまとめる7個の2分木ノードが全てあるが、その内のいくつかには経路情報が割り付けられていない例である。最下の4つのノードの内、経路情報が割り付けられていないノードA01、A10には、そのノードの上方につながっているノードの内、経路情報が割り付けられている最も下、即ち最もマスク長が長いノード（それぞれ、A、A1）の経路情報 * A、* a1を入れる。

【0065】図31（b）は一つにまとめる7個の2分木ノードの内のいくつかしかノードが存在しない例であり、存在しないノードA01、A10をまず経路が割り付けられていないノードとして補い、図31（a）と同じ規則で経路情報を入れる。最下の4つのノードA00、A01、A10、A11の内、補ったノードA01、A10の下にはノードが繋がっていないので、下のノードへのポインタにはNULLを入れる。

【0066】一般に2のp乗分木の場合も同様にして、一つにまとめる2のp乗-1個の2分木ノードをつぶして、一番下の2の（p-1）乗個のノードだけの大きさにする。

【0067】この方法で1ノードの大きさを小さくすることは、メモリ効率の悪化を防ぐ効果がある。試みに、以下に示す近似の下で木を2分木で作成した場合と2の

p乗分木で作成した場合のメモリ使用量を計算し、2のp乗分木にした場合でも、pが小さい場合には、メモリ効率が悪くならないことを示す。

【0068】近似

1. 2分木にした場合、経路は全て末端のノードにのみ割り当てられている。

【0069】2. 木の枝は第32bitまで全てある。

【0070】3. 枝の増え方は、全bitで一定とする。

【0071】4. 経路数は約10k経路とする。

【0072】この近似の下で構成した木の形を図32に示す。この近似によると、1bit下る毎の木の広がり方は10kの（1/32）乗、即ち約1.33倍である。

【0073】この構成の木を、2分木ノードで構成した場合と4分木ノードで構成した場合のメモリ量の比較を、図33（a）を用いて説明する。4分木ノードを構成する3つの2分木ノードの内、上の一つのノードN2に着目すると、直下にはノードが約1.33個付く。即ち、左右各々0.67の確率でノードN20、N21が付く。したがって、4分木ノードを構成する2分木ノードN2、N20、N21の合計のメモリ量のノードの存在確率を考慮した平均は、2分木ノード（1+1.33）個分、即ち2.33個分である。4分木ノードで木を構成すると、この3つの2分木ノードが一つの4分木ノードN4になり、この一つの4分木ノードのメモリ使用量は2分木ノードN20、N21の2個分である。

同様にして、この構成の木を、2分木ノードで構成した場合と8分木ノードで構成した場合のメモリ量の比較を、図33（b）を用いて説明する。8分木ノードを構成する7つの2分木ノードN2、N20、N21、N200、N201、N210、N211の内、最上の一つのノードN2に着目すると、直下にはノードN20、N21が約1.33個付き、そのさらに下にはノードN200、N201、N210、N211が約1.33の2乗、即ち約1.78個付く。したがって、8分木ノードを構成する2分木ノードN2、N20、N21、N200、N201、N210、N211の合計のメモリ量のノードの存在確率を考慮した平均は、2分木ノード（1+1.33+1.78）個分、即ち4.11個分である。

【0074】8分木ノードで木を構成すると、この7個の2分木ノードが一つの8分木ノードN8になり、この一つの8分木ノードのメモリ使用量は2分木ノードN200、N201、N210、N211の4個分である。

【0075】同様にして2分木の場合と4、8、16、32、64、128、256分木の場合のメモリ容量の比較を行った結果を以下に示す。

【0076】

2分木: 4分木= 2. 33: 2=0. 86倍
 2分木: 8分木= 4. 11: 4=0. 97倍
 2分木: 16分木= 6. 48: 8=1. 23倍
 2分木: 32分木= 9. 65: 16=1. 66倍
 2分木: 64分木=13. 86: 32=2. 31倍
 2分木: 128分木=19. 49: 64=3. 28倍
 2分木: 256分木=26. 98: 128=4. 74倍
 経路数が1M経路の場合には、この近似によると、1bit下る毎の木の広がり方は1Mの(1/32)乗、即ち約1. 54倍になり、2分木の場合と4、8、16、32、64、128、256分木の場合のメモリ量の比は、以下の通りとなる。

【0077】

2分木: 4分木= 2. 54: 2=0. 79倍
 2分木: 8分木= 4. 91: 4=0. 81倍

3個の2分木が、2分木のノードの2倍の大きさの4分木になる。

【0079】

7	"
15	"
31	"
63	"
127	"
255	"

4	"	8	"
8	"	16	"
16	"	32	"
32	"	64	"
64	"	128	"
128	"	256	"

(2) 1bit下る毎の木の広がりが大きく、1ノード内のデータ使用効率が良い。(サポート経路数が増える程、木の広がりが大きくなるので、1ノード内のデータ使用効率は良くなる。)

さらに、4、8、16、…分木ノードでは、2分木ノードを2、4、8、…個まとめて扱うので、一つにまとめられる2分木ノード間で一つだけ持てば良い要素は一つだけ持てばよく、これにより4、8、16、…分木ノードのメモリ量をさらに小さくできる。一つにまとめられる2分木ノード間で一つだけ持てば良い要素には、サブネットワークアドレス及びサブネットワークマスク長があるが、サブネットワークマスク長については後述するようにこのノード自身のサブネットワークマスク長ではなく、このノードの直下のノードのサブネットワークマスク長を持つようにするので、メモリ量を小さくする効果は無い。

【0080】2分木ノードの構造、及びこの2分木ノードを2つ併せ、2分木間で一つだけ持てば良いサブネットワークアドレスを一つだけ持つようにした場合の4分木ノードの構造を各々図34、図35に示す。図34は2分木ノードの構造を示す図であり、次のノードのマスク長0、1は、このノード自身のサブネットワークマスク長ではなく、このノードの直下のノードのサブネットワークマスク長である。このように、自分自身でなく直下のノードのマスク長を設定する理由は高速化のためであり、その説明は図36を参照して後述する。Flag0及びFlag1は、このノードに対応するサブネット

2分木: 16分木= 8. 56: 8=0
 2分木: 32分木=14. 19: 16=1
 2分木: 64分木=22. 85: 32=1
 2分木: 128分木=36. 18: 64=1
 2分木: 256分木=56. 72: 128=2
 結論として、上記仮定の下では10k経路時まで、1M経路時には16分木までなら、使用効率は良くなる。256分木にした場合使用量は10k経路時に3. 28倍、1M2. 26倍までしか増えない。このようにノ率があまり悪くならない理由として、以下のられる。

【0078】(1) p段分のノードを纏め、纏める前のノードの合計よりも、纏めたがコンパクトになる。即ち、

3個の2分木が、2分木のノードの2倍の大きさの4分木になる。

ワークがこのルータに直接つながるか、他のつ以上経由してつながるかを示すビット、及を使用し後述するように、このノードが経路られているノードか否か、即ち、図8に示しはこのノードが“*”が付いているノードかフラグ、他である。Flag0とFlag1値を入れる。これは、ワードW0とワードW1を讀めば良いようにするためである。このワードの全てを讀むのではなく、一部分を讀む高速化については図36を使用し後述する。へのポインタ0、1は、宛先アドレスのこのマスク長で示されるビット位置の値が、それぞれに次に辿るノードへのポインタである。ワークアドレスは、このノードに対応するネットワークアドレスである。出力ポート番号及びアドレスは、このノードに割り付けられた経路情報入ってきたパケットを出力すべきポート及上のパケットを送るべきルータのアドレス

【0081】4分木ノードでは、この2分木を2つ併せ、併せたときに一つだけ持てば良いように持つようにする。一つだけ持てば良いネットワークアドレスだけである。この方が4分木ノードの構造を図35に示す。

【0082】図34、図35に示す例では、ドは、2のべき乗の大きさである16バイト入りきらない大きさになっているが、4分木し、1ノード内にサブネットワークアドレス

9 3 倍
1 3 倍
4 0 倍
7 7 倍
2 6 倍
は 8 分木
る メモリ
も、メモ
路時に
り使用効
率が挙げ

とによ
のノード

ータを一
、図 3 7
割り付け
木の例で
かを示す
は、同じ
の一つだ
うに、ノ
とによる
次のノード
ノードのマ
0、1 の
サブネット
ネットワーク
ホップアド
最であり、
そのポート
ある。
ノードを
ータを
ータに
で作

2

2 のべき乗の大きさである
になっている。8 分木ノ
レスを 1 ノードで一つだ
トの大きさに収まった上
の領域は他の情報を入れ
にまとめる 2 分木の数を
に対し、一つのノードの
ードの大きさを 2 のべき
w の構成を非常に簡単
こでできる例とその利点を

4 分木ノードが 3 2 バ
バンクで構成していた
バンク境界にまたがる
ミック RAM を使用し
領域が Row アドレス
利点として挙げられ

ノード内の各要素のア
ドへのポインタとその
算でなく、アドレスの
下位ビットをオフセッ
木ノードが 3 2 バイト
要素のアドレスは、
スの 2 の 5 乗ビット以
へのオフセットをア
1 乗ビットに割り付け
る。

例えば 4 分木ノードが
に保持する次のノード
の先頭のバイトアド
ノード内で 1 ポインタ
減らせることが利点とし

1 6、... 分木ノードにし
なり、検索処理時に検索
全て読み込むと、ノード
時間が伸び、性能低下要因
この問題は、ノードを大き
てを読み込まずに一部だけ
避する。この方法につい
る。

この場合の例であり、既に図
にマスク長 m ビットの 4 分木
m ビット目の値が 0 の場合に
1 の場合に対応する 2 分木ノ
ることから、宛先アドレスの
対応する方の 2 分木ノードの部
なり、ノードの大きさが大きく

なっても 2 分木ノードの場合と同じデータ量を読み込む
ようにする。このとき、図 3 5 で示した、一つにまとめ
られる 2 分木ノード間で一つだけ持つ要素であるサブネ
ットワークアドレスは、宛先アドレスの m ビット目の値
に係わらず読み込むようにする。

【0089】さらに、宛先アドレスの m+1 ビット目の
値を見て、2 分木ノードで 2 つ存在した次ノードへのポ
インタの内、一方だけを読み込むようにすることによ
り、読み込むデータ量をさらに少なくする。

【0090】この方法は 2 分木の場合でも使用できる。
例えば m ビット目の 2 分木の場合には、宛先アドレスの
m ビット目の値を見て、2 つの次ノードへのポインタの
内、一方だけを読み込むようにする。

【0091】上記方法を全て行い、結局、このノードの
マスク長を m とした場合、宛先アドレスの第 m、m+1
ビットの値が 0 0 か、0 1 か、1 0 か 1 1 かに従い、そ
れぞれ (W0→W4→W5→W6)、(W1→W4→W5→W6)、
(W2→W4→W5→W7)、(W3→W4→W5→W7) の順にデータを読み込むようにする。

【0092】このように、あるノードの一部分だけを読
み込むためにはこのノードのマスク長 m を知る必要があ
り、このノードのマスク長 m は 1 ノードのデータ読み込
みの最初に読み込むか、この情報の一つ前のノードに移
して一つ前のノードのデータ読み込み時に読み込む必要
がある。ノードのマスク長 m を 1 ノードのデータ読み込
みの最初に読み込む方法は、宛先の第 m ビット目の値の
抽出のための検索処理 L S I 内のゲートディレイ、及
び、次に読み込む部分のアドレスをメモリに出力してから
メモリからのデータを検索処理 L S I 内に読み込むま
での時間であるメモリリードレイテンシだけ、マスク長
m を読み込んでから次に読み込む部分を選択して読み込
むまで時間が空いてしまうので、ノードの一部だけを読
むことによる性能向上効果が少なく、ノードのマスク長
m を一つ前のノードに移し、一つ前のノードのデータの
読み込み時に読み込む方が性能向上効果がある。

【0093】さらに、ノードのマスク長 m を一つ前のノ
ードに移す場合、1 ノードのデータを読み込む順序を、
1 番目に次のノードのマスク長 m、次のノードへのポイ
ンタ、次にサブネットワークアドレス、出力ポート番
号、及び、次ホップアドレスの順にすることにより、次
のノードの最初に読み込む部分のアドレスが最も早く計
算できるようにする。

【0094】次のノードへのポインタは、次のノードの
メモリ領域の先頭部分を指しており、次のノードの先頭
から最初に読み込む部分までのアドレスのオフセット
は、次のノードのマスク長 m を読み込み、宛先アドレス
の該当ビット位置の値を検査することにより、得られ
る。

【0095】次に、1 ノード内で、条件によっては、読
み込む必要が無い要素を、条件に従い、読み込まないよ

うにすることで、読み込みの時間を削減することで、高速化を図る方法について図37を使用し説明する。

【0096】図37は4分木の場合の例である。Radishアルゴリズムでは、全ノードに経路が割り付けられているわけではなく、枝の分岐の個所では、経路が割り付けられていなくてもノードを設ける必要がある。図37に示すように、ノードデータの最初に読み込むフラグ内に、このノードが経路が割り付けられているノードか否かの情報を入れておき、経路が割り付けられていないノードでは、出力ポート、及び、ネクストホップアドレスを読み込まないようにすることで、読み込み時間の短縮が図れる。このノードが経路が割り付けられているノードか否かの情報は、1ビットで表現できるので、この情報を読み込むことによる読み込み時間の増大は小さい。

【0097】この方法では、このノードのマスク長を m とすると、宛先アドレスの第 m 、 $m+1$ ビットの値が00で、 $W0$ を読み、Flag00から、4分木を構成する0番目の2分木に経路情報が無いと判った場合、 $W4$ だけを読めば良く、経路情報が有ると判った場合にだけ、図36に示すように $W4 \rightarrow W5 \rightarrow W6$ の順に読めば良い。宛先アドレスの第 m 、 $m+1$ ビットの値が01、10、11の場合も同様である。

【0098】次にここまでで示した方式の経路検索を行い、経路検索の結果得られた宛先にパケットを転送するルーティングプロセッサ(図39の210)の内部構成例について図40を用いて説明する。

【0099】図40において、ルーティングプロセッサ210は、パケット転送処理を行う転送処理部211と、転送処理部211が、ルーティングプロセッサに入ってきたパケットを転送するまでの間、一時的にパケットデータを格納しておくパケットバッファメモリ212と、パケットのヘッダ情報に基づき経路検索を行う経路検索処理部213と、経路検索処理部213が検索する経路テーブルを格納する経路テーブルメモリ214を有する。

【0100】次に、ルーティングプロセッサ210におけるパケットの中継処理の動作を説明する。なお、パケット中継処理を行う前に、ルータ管理部(図39の260)が、ルータ装置内通信手段220に接続しているルーティングプロセッサ210に、それぞれの経路テーブルを配布しており、各ルーティングプロセッサ210は経路テーブルメモリ214に経路テーブルを格納している状態とする。

【0101】転送処理部211はパケットバッファメモリ212に格納されたパケットデータの内、パケットのヘッダ情報を抽出して経路検索処理部213へ渡し、経路検索処理部213は、受信パケットのヘッダ内の宛先アドレスを用いて経路テーブルメモリ214内に格納されている経路テーブルの検索を行い、検索結果として、

経路情報を転送処理部211へ渡す。経路情報は、経路が存在するか否かのFlag、図34乃至図37に示すFlag、次転送先アドレス、出力ルーティングプロセッサ(図39の210)の番号、及び、出力ポート(図39の240)の番号である。転送処理部211は、経路検索処理部213から渡された経路情報に従い、出力先が自ルーティングプロセッサ210に繋がるポート

(図39の240)の場合、ネットワークインタフェース部(図39の230)にパケットを転送し、他ルーティングプロセッサ(図39の210)に繋がるポート

(図39の240)の場合、ルータ装置内通信手段(図39の220)にパケットを転送する。

【0102】以上、2分木検索方式(以下方式1と呼ぶ)、2の p 乗分木検索方式(以下方式2と呼ぶ)、およびマスク長 m ビットのノードを経路テーブルメモリ214上に展開する方式(以下方式3と呼ぶ)という各検索方式について説明した。次に、図40の経路検索処理回路200が上記の方式を用いて経路検索処理を行う際のフローチャートについて図41を用いて説明する。図41のフローチャートでは、方式2と方式3とを組み合わせた場合の例を示す。この例では、経路アドレスの第0ビットから第 $(m-1)$ ビットの値に従って、経路テーブルメモリ214上の決まった位置に展開されたノードの一つを選択する。以下ではこのように選択し、検索の最初に読込むノードを初段ノードと呼ぶ。第 m ビット以降は経路アドレスを p ビットずつ検索し、2の p 乗分木を検索する。図40の経路テーブルメモリ214には、上記の方式2と方式3の検索方式に従った2の p 乗分木ノードデータ、および、次転送先アドレス、出力ルーティングプロセッサ210の番号、及び、出力ポート240の番号が格納されているとする。

【0103】なお、以下では方式2と方式3とを組み合わせた例について説明するが、方式1と方式3とを組み合わせた場合でも同様なフローチャートに従うことで実現可能である。また、このフローチャートに従うことにより、ソフトウェアでもハードウェアでも経路検索処理を実現することができる。ソフトウェアで実現する場合、図40の経路検索処理部213にはCPUを用いればよい。またハードウェアで実現する場合、図40の経路検索処理部213を専用LSIで構成すればよい。

【0104】図41の処理810は木構造検索処理であり、処理811は経路情報出力処理である。まず、木構造検索処理810について説明する。

【0105】図40の経路検索処理部213は、受信パケットの宛先IPアドレスと初段ノードのマスク長 m の値から初段ノードへのポインタを生成し、このポインタと、宛先IPアドレスの第 m ビットから第 $(m+p-1)$ ビットの値(以下、検査ビット値と呼ぶ)に従って経路テーブルメモリ214に格納されている初段ノードの読み込みアドレスを生成し、経路テーブルメモリ214から

該初段ノードの一部を読み込む(図41の800)。

【0106】次に、図40の経路検索処理部213は、受信パケットの宛先IPアドレスにノードのマスク長だけ上位ビットから有効とするマスクをかけたものと、ノードの経路アドレスとを比較し(図41の801)、不一致の場合は木構造検索処理(図41の810)を終了する(図41の809)。一致する場合は図41の処理802に進む。図41の処理802、および803は、最長一致検索を実現するための経路情報の更新処理である。経路情報は、図34乃至図37に示すFlag、次転送先アドレス、出力ルーティングプロセッサ(図39の210)の番号、及び、出力ポート(図39の240)の番号である。図40の経路検索処理部213は、ノードデータの内のFlag中のエントリ有りフラグを検査し、エントリ有りフラグの値が1の場合(図41の812)のみ、読み込んだノード内の新たな経路情報をレジスタに保持する(図41の803)。エントリ有りフラグの値が0の場合は更新処理を行わない(図41の813)。

【0107】次に、図40の経路検索処理部213は、図40の経路テーブルメモリ214が出力するノードデータの内の次ノードへのポインタがNULLかどうかを判定し、NULLの場合は木構造検索処理810を終了する。NULLでない場合はそのポインタと、新たな検査ビットの値に従い経路テーブルメモリ214に格納されている初段ノードの読み込みアドレスを生成し、経路テーブルメモリ214から該ノードデータを読み込む(図41の805)。

【0108】上記の処理を繰り返すことにより、2のp乗分木方式の経路検索を行うことができる。

【0109】次に、図41の経路情報出力処理811について説明する。木構造検索処理の結果、図40の経路検索処理部213内に経路情報が保持されている。図40の経路検索処理部213は、まず上記のエントリ有りフラグを調べ(図41の806)、その値が0の場合は経路検索処理を終了し、転送処理部211へ検索結果無しという通知をする。エントリ有りフラグの値が1の場合は、検索の結果、あるエントリに一致したことになるため、経路情報を転送処理部211へ出力する。

【0110】次に、本発明の一実施例として、図41で説明した検索方式をハードウェアで実現する場合の構成例を図42、図43を用いて説明する。

【0111】図42に経路検索処理部213をハードウェアで構成した場合の構成例を示す。経路検索処理部213は、木構造検索回路2130と、読み込みアドレス生成回路2131と、経路検索処理制御回路2132とからなる。

【0112】木構造検索回路2130は、経路テーブルメモリ214に格納された2のp乗分木構造を検索し、次に読み込むべきノードのポインタの生成、受信パケッ

トの宛先IPアドレスの検査ビット値の抽出、木構造検索の終了判定、検索結果である経路情報の候補の更新を行う。また、読み込みアドレス生成回路2131は、木構造検索回路2130から出力される読み込むべきノードへのポインタ、および検査ビット値に従い、実際に読み込むノードの一部のワードのメモリアドレスを生成する。また、経路検索処理制御回路2132は、経路検索処理部213全体の制御(各回路の動作タイミングおよび動作状態管理など)を行う。

【0113】次に、経路検索処理部213の動作について図42を用いて説明する。また、木構造検索回路2130の詳細動作については、図43を用いて後述する。

【0114】木構造検索回路2130は、転送処理部211から受信パケットの宛先IPアドレスを受け取り、この宛先IPアドレスとノードのマスク長の値から次ノードへのポインタを生成して、読み込みアドレス生成回路2131に渡す。また、木構造検索回路2130は、ノードのマスク長で示される宛先IPアドレスの検査ビット位置の値(検査ビット値)を抽出して、読み込みアドレス生成回路2131に渡す。

【0115】読み込みアドレス生成回路2131はこのノードへのポインタと、検査ビット値と、経路検索処理制御回路2132からのタイミング信号を用いて、読み出すべきノードデータが格納されているメモリアドレスを生成し、メモリ制御回路2132へ送信し、メモリ制御回路2132は上記メモリアドレスと経路検索処理制御回路2132からのタイミング信号を用いてメモリ制御信号を生成し、経路テーブルメモリ214へ転送する。上記のメモリ制御信号を受信した経路テーブルメモリ214は、対応するノードデータを信号線215を用いて木構造検索回路へ転送する。

【0116】木構造検索回路2130はこのノードデータを用いて、図41の処理801、802、803、804、805を行う。これらの処理の詳細は図43で後述する。図41の処理801および804に対応する判定処理において木構造検索を終了すると判定した場合は、木構造検索終了信号を経路検索処理制御回路2132へ出力し、経路検索処理制御回路2132は、木構造検索回路2130内に保持された経路情報のうちのエントリ有りフラグを調べ、その値が0の場合は経路検索処理を終了し、転送処理部211へ検索結果無しという通知をする。エントリ有りフラグの値が1の場合は、経路情報を出力し検索処理を終了し、次のパケット処理の制御を行う。

【0117】次に、図42の木構造検索回路2130の詳細を図43を用いて説明する。

【0118】まず、図41の初段ノードリード処理800および次ノードリード処理805に対応する処理について、図43を用いて説明する。転送処理部211より信号線216を用いて渡される宛先IPアドレスは、宛

先IPアドレスバッファ213001内に保持され、初段ノードへのポインタ生成回路213002、検査ビット抽出回路213006、マスク処理回路213010に入力される。

【0119】初段ノードへのポインタ生成回路213002は、予め図39のルータ管理部260から初段ノードマスク長レジスタ213010に設定された値 m に従い、受信パケットの経路アドレスの第0ビットから第 $(m-1)$ の上位 m ビットの値を抽出し、この m ビットの値に従って初段ノードへのポインタを生成してアドレスセクタ213003に出力する。アドレスセクタ213003は、信号線2134によって図42の経路検索処理制御回路2132から出力される初段ノード読み込み/初段ノード以外のノード読み込み選択信号に従い、初段ノードの読み込み時は、上記の初段ノードへのポインタを選択して図42の読み込みアドレス生成回路2131に出力する。また、アドレスセクタ213003は、初段ノード以外のノードの読み込み時は、次ノードへのポインタバッファ213004に保持されている次ノードへのポインタを選択して読み込みアドレス生成回路2131に出力する。

【0120】また、上記の処理と並行して、検査ビット抽出回路213006は、初段ノード読み込み時には、検査ビット位置セクタ213005において選択されて出力される初段ノードマスク長レジスタ213000の設定値 m に従い、経路アドレスの第 m ビットから第 $(m+p-1)$ ビットまでの p ビットの検査ビット値を抽出して図42の読み込みアドレス生成回路2131へ出力する。また、初段ノード以外のノードの読み込み時は、後述する次ノードマスク長バッファ213007に保持されている次ノードのマスク長 $m1$ が検査ビット位置セクタ213005において選択され、この次ノードのマスク長 $m1$ が検査ビット抽出回路213006に入力され、検査ビット抽出回路213006は、この次ノードのマスク長 $m1$ に従い、経路アドレスの第 $m1$ ビットから第 $(m1+p-1)$ ビットまでの p ビットの検査ビット値を抽出して図42の読み込みアドレス生成回路2131へ出力する。

【0121】図42の読み込みアドレス生成回路2131およびメモリ制御回路2133は、上記のノードへのポインタと p ビットの検査ビット値を用いて、図36乃至図37に示した順にノード内の各ワードのアドレスを生成し、経路テーブルメモリ214は入力されたメモリアドレスに格納されているノードのデータを信号線215に出力する。

【0122】信号線215のビット幅が32ビットの場合、図36で説明した方式を採用するとノードデータの一回の読み込みワード数は4ワードとなり、これらのデータは図36の表に示した順番で信号線215に出力され、各バッファ(213004、213007、213

013、213014、213015)に保持される。各バッファの保持タイミングは経路検索処理制御回路2132からの制御信号(図示していない)により制御される。バッファ213004には次ノードへのポインタが保持され、バッファ213007には次ノードのマスク長が保持され、バッファ213013にはノードのサブネットワークアドレスが保持され、バッファ213014にはフラグが保持される。

【0123】次に、図41の経路アドレス一致比較処理801に対応する処理について、図43を用いて説明する。初段ノード読み込み時には、マスク長セクタ213009を介して初段ノードマスク長レジスタ213000の設定値がマスク処理回路213010に入力される。初段ノード以外のノード読み込み時には、マスク長セクタ213009を介してマスク長バッファ213008に保持されているノードのマスク長が入力される。このマスク長バッファ213008の値は、一つ前に読み込んだノードに格納されている次ノードのマスク長の値であり、次ノードマスク長バッファ213007に保持されていたものである。次ノードマスク長バッファ213007の値は、ノードの読み込み毎に更新されるため、更新される前に、現在読み込んでいるノード(以下、現ノードと呼ぶ)の経路アドレス一致比較処理に使用する現ノードのマスク長をマスク長バッファ213008に保持しておく。

【0124】マスク処理回路213010は、これらのマスク長だけ上位ビットから有効とするマスクを生成し、このマスクと、宛先IPアドレスバッファ213001から出力される宛先IPアドレスとの論理積をとり、その結果(以下、マスクした受信パケットの宛先IPアドレスと呼ぶ)が一致比較回路213011に入力される。また、サブネットワークアドレスバッファ213013に保持されているノードのサブネットワークアドレスも一致比較回路213011に入力される。一致比較回路213011は、このノードのサブネットワークアドレスと、上記で説明した、マスクされた受信パケットの宛先IPアドレスとを比較し、その結果が不一致となる場合に不一致信号を信号線213019を用いて木検索終了判定回路213012に出力する。木検索終了判定回路213012は上記不一致信号を入力し、木構造検索終了信号を図42の経路検索処理制御回路2132に出力する。

【0125】また、マスク長バッファ213008の値は、上記の経路アドレス比較が行われた後、次ノードマスク長バッファ213007に保持されている次ノードのマスク長の値によって更新され、次のノードの読み込み時のサブネットワークアドレス比較に使用される。

【0126】次に、図41の処理802、および803に対応する処理について、図43を用いて説明する。

【0127】一致比較回路213011におけるサブネ

ットワークアドレス比較の結果、一致した場合、一致比較回路213011は信号線213020を用いて一致信号を更新判定回路213018に出力する。この一致信号が更新判定回路213018に入力され、かつ、フラグバッファ213014に保持されているフラグの内のエントリ有りフラグの値が1の場合(図41の812)のみ、更新判定回路213018は更新信号をフラグ候補バッファ213016および経路情報候補バッファ213017に出力する。更新信号を受信したフラグ候補バッファ213016は、フラグバッファ213014に保持されているフラグを新たに保持し、同じく更新信号を受信した経路情報候補バッファ213017は、経路情報バッファ213015に保持されている経路情報を新たに保持する(図41の経路情報更新処理803)。エントリ有りフラグの値が0の場合は更新判定回路213018は更新信号を送信しないので、フラグ候補バッファ213016および経路情報候補バッファ213017は更新処理を行わない(図41の813)。

【0128】経路情報として上記のフラグ、及び、経路情報以外の情報を追加する必要がある場合は、木構造のノード内にそれらの情報を追加し、それらの情報を保持、更新するバッファを新たに追加すればよい。

【0129】次に、図41の次ノードへのポインタがNULLかどうかの判定処理804に対応する処理について、図43を用いて説明する。次ノードへのポインタバッファ213004内に保持されている次ノードへのポインタは木検索終了判定回路213012に入力される。この次ノードへのポインタがNULLの場合、木検索終了判定回路213012は図42の経路検索処理制御回路2132に木検索終了信号を出力する。

【0130】以上、図42の木構造検索処理回路2130の動作を説明したが、ハードウェアで構成するため、図41の木構造検索処理の中の801、802、803、804、805の各処理は逐次処理をする必要はなく、各処理に必要なデータが各バッファ213004、213007、213013、213014、213015に保持された後に、各処理を開始すればよく、上記の各処理を並列処理を行うことにより、高速に木構造の検索を行うことができる。

【0131】

【発明の効果】前述の説明の通り、本発明はルータ等のネットワーク中継装置に用いて好適なネットワーク次転送先検索方法及びそれを用いたネットワーク中継装置であり、ネットワーク中継装置が受信したパケットの転送先アドレスを高速に検索し、ネットワーク中継装置のパケット処理性能を向上させることができる。

【図面の簡単な説明】

【図1】本発明が前提とする一般的なネットワークシステムの構成図である。

【図2】ルータにおける経路検索仕様を説明する図であ

る。

【図3】アドレス長3ビットの場合の全てのノードがある2分木を説明する図である。

【図4】アドレス長3ビットの場合の経路テーブル例を示す図である。

【図5】経路が割り付けられておらず、かつ、経路付きのノードへの途中経路にもなっていないノードを取り除いた木を説明する図である。

【図6】アドレス長32ビットの場合の経路テーブル例を示す図である。

【図7】図6に示した経路テーブルに対応する木を説明する図である。

【図8】枝別れも経路の割り付けもないノードを取り除いた木を説明する図である。

【図9】図9はマスク長2ビットのノードをメモリ上に展開し第0～第1ビットの検索時間を除いた木を説明する図である。

【図10】マスク長kビット目までのノードを経路検索LSI内に入れた場合のメモリ構成図である。

【図11】従来のパイプライン処理を行わない経路検索処理のタイムチャートを示す図である。

【図12】マスク長kビット目までのノードを経路検索LSI内に入れた場合の経路検索のパイプライン処理を表すタイムチャートを示す図である。

【図13】2分木から8分木への変形時に一つの8分木ノードにまとめられる2分木ノードを囲んだ木を説明する図である。

【図14】2分木から8分木への変形時に一つの8分木ノードにまとめられる2分木ノードを囲んだ木を説明する図である。

【図15】2分木から8分木への変形時に一つの8分木ノードにまとめられる2分木ノードを囲んだ木を説明する図である。

【図16】マスク長が0ビットから始まらないようにビット位置を区切った場合に最初のノードをメモリ上に展開することにより区切り位置までのビットの検索を行う木を説明する図である。

【図17】マスク長が0ビットから始まらないようにビット位置を区切った場合に最初のノードをメモリ上に展開することにより区切り位置までのビットの検索を行う木を説明する図である。

【図18】図9と組み合わせ、先頭のさらに多くのビット数の検索時間を除いた木を説明する図である。

【図19】図16と組み合わせ、先頭のさらに多くのビット数の検索時間を除いた木を説明する図である。

【図20】図17と組み合わせ、先頭のさらに多くのビット数の検索時間を除いた木を説明する図である。

【図21】2分木から4分木への変形時に一つの4分木ノードにまとめられる3つの2分木ノードを2つの2分木ノード分につぶした木を説明する図である。

【図22】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図23】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図24】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図25】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図26】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図27】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図28】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図29】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図30】一つの4分木ノードにまとめられる3つの2分木ノードの2つの2分木ノードへのつづし方を説明する図である。

【図31】一つの8分木ノードにまとめられる7つの2分木ノードの4つの2分木ノードへのつづし方を説明する図である。

【図32】木の構成に必要なメモリ量見積り時の木の広がりモデルを説明する図である。

【図33】木を2分木ノードで構成した場合と4、8分木ノードノードで構成した場合のノードの存在確率を考慮したメモリ量の比較を示す図である。

【図34】2分木ノードの構造を示す図である。

【図35】4分木ノードの構造を示す図である。

【図36】図36はノードの大きさが大きくなったときにノードデータリード時間の増大を防ぐために一つのノード全てを読み込まずに一部だけを読み込む方法を示す図である。

【図37】条件によっては読み込む必要が無い要素を、条件に従い読み込まないようにすることで読み込みの時

間を削減することで高速化を図る方法を示す図である。

【図38】ルータ装置の一構成例を示すブロック構成図である。

【図39】ルータ装置の他の構成例を示すブロック図である。

【図40】本発明の一実施例であるルータ装置内ルーティングプロセッサのブロック図である。

【図41】本発明の一実施例である経路検索処理部の検索処理フローチャートである。

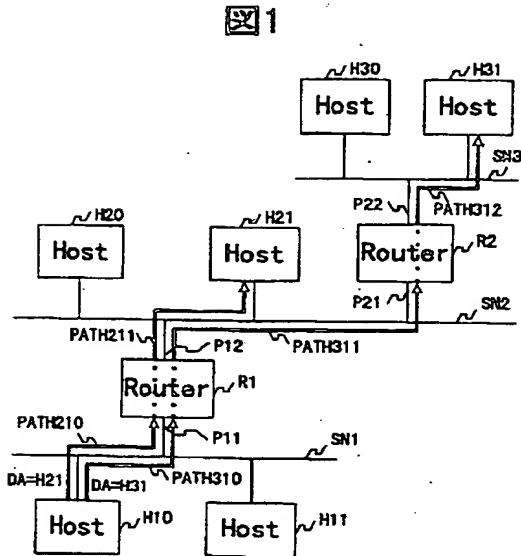
【図42】本発明の一実施例である経路検索処理部のブロック図

【図43】本発明の一実施例である、木構造の経路検索を行う木構造検索処理回路のブロック図である。

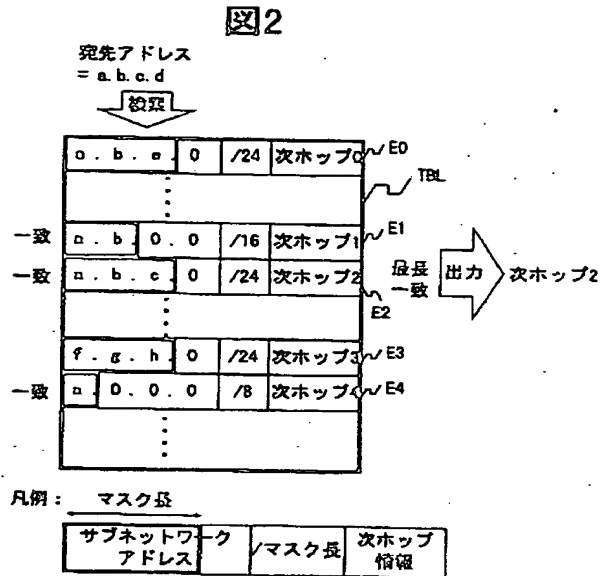
【符号の説明】

100、200…ルータ、110…ルーティング制御部、120…ルータバス、130…ネットワークインタフェース部、140…ポート、150…サブネットワーク、220…ルータ装置内通信手段、230…ネットワークインタフェース部、240…ポート、250…サブネットワーク、260…ルータ管理部、211…転送処理部、212…パケットバッファメモリ、213…経路検索処理部、214…経路テーブルメモリ、215…経路テーブルメモリリードデータ、216…宛先IPアドレス、810…木構造検索処理、811…経路情報出力処理、2130…木構造検索回路、2131…読み込みアドレス生成回路、2132…経路検索処理制御回路、2133…メモリ制御回路、2134…タイミング制御信号、213000…初段ノードマスク長レジスタ、213001…宛先IPアドレスバッファ、213002…初段ノードへのポインタ生成回路、213003…アドレスセクタ、213004…次ノードへのポインタバッファ、213005…検査ビット位置セクタ、213006…検査ビット抽出回路、213007…次ノードマスク長バッファ(m1)、213008…マスク長バッファ、213009…マスク長セクタ、213010…マスク処理回路、213011…一致比較回路、213012…木検索終了判定回路、213013…サブネットワークアドレスバッファ、213014…フラグバッファ、213015…経路情報バッファ、213016…フラグ候補バッファ、213017…経路情報候補バッファ、213018…更新判定回路、213019…不一致出力信号、213020…一致出力信号。

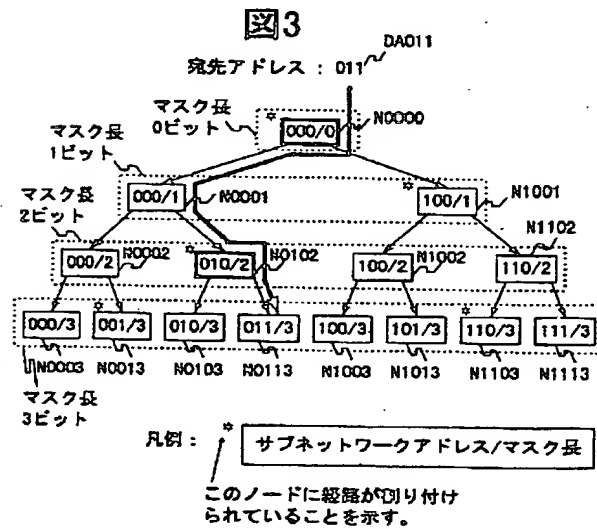
【図1】



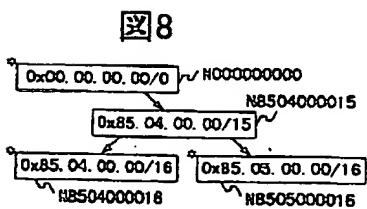
【図2】



【図3】



【図8】



【図4】

図4は、経路テーブルを示す。

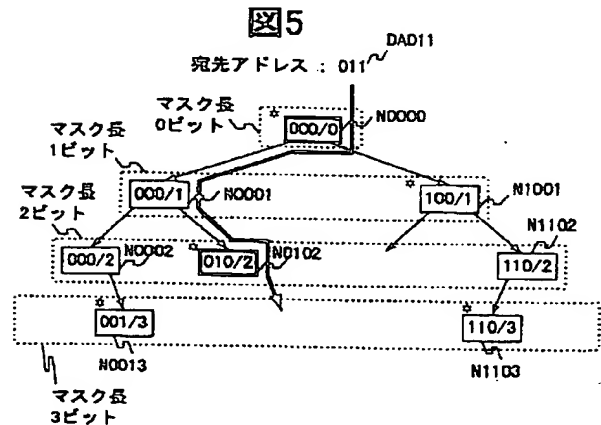
サブネットワーク アドレス	マスク 長
000	0
001	3
010	2
100	1
110	3

【図6】

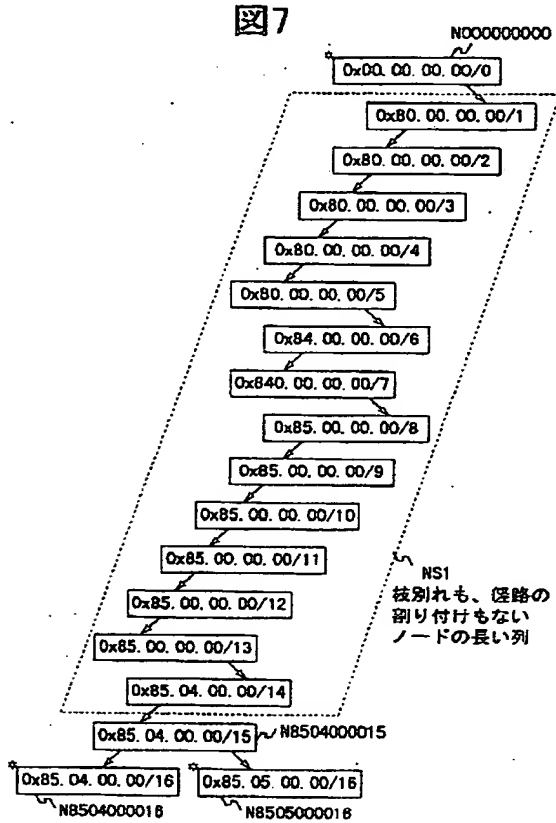
図6は、経路テーブルを示す。

サブネットワーク アドレス	マスク 長
0x00.00.00.00	0
0x85.04.00.00	18
0x85.05.00.00	16

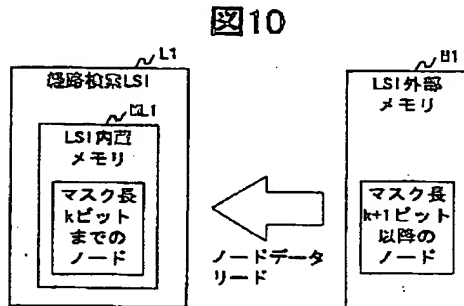
【図5】



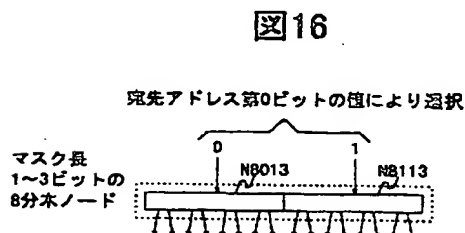
【図7】



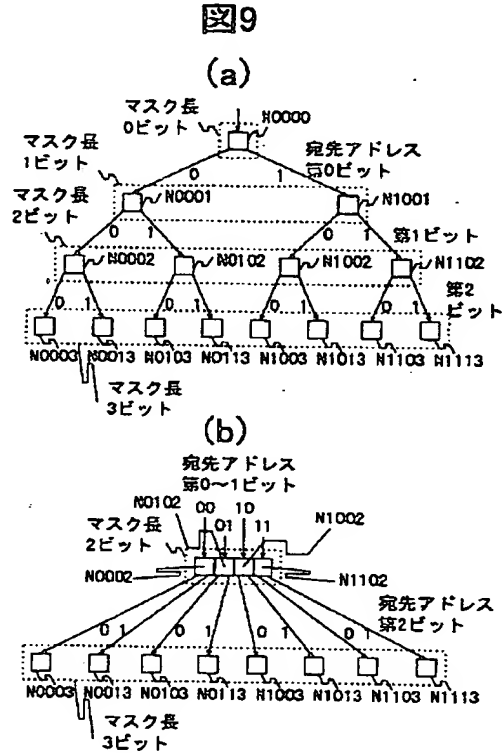
【図10】



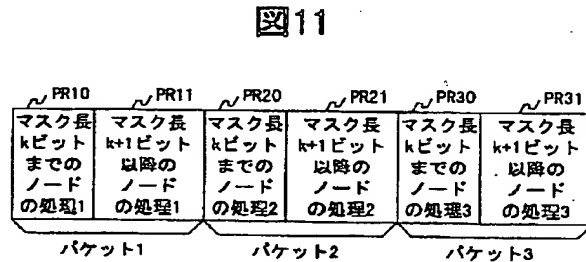
【図16】



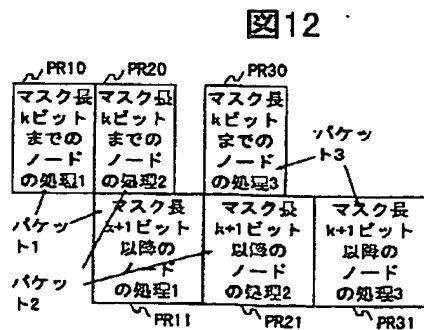
【図9】



【図11】

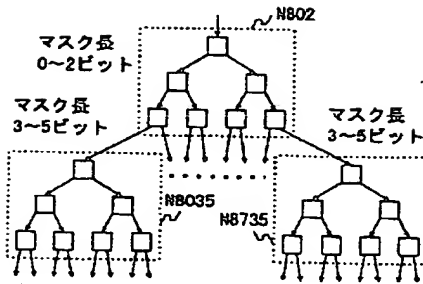


【図12】



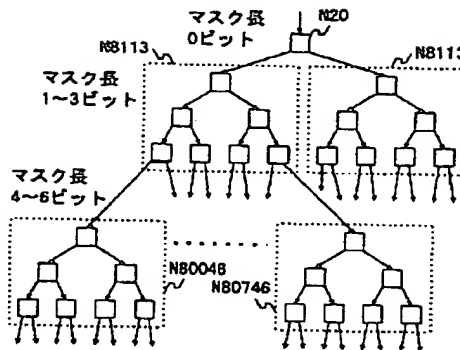
【図13】

図13



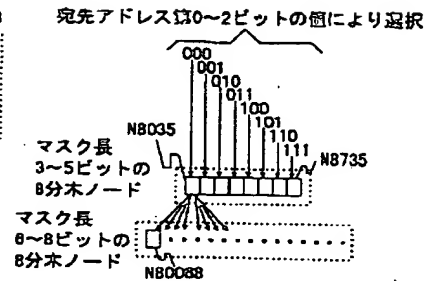
【図14】

図14



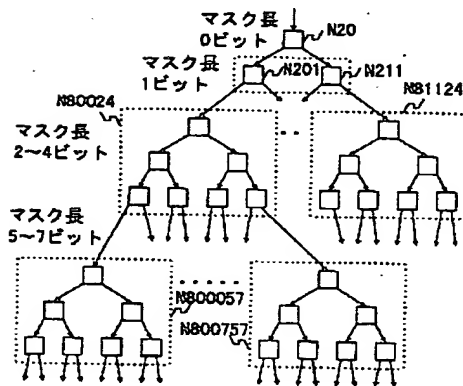
【図18】

図18



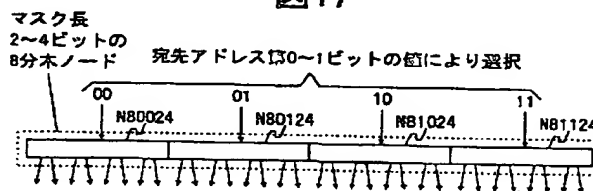
【図15】

図15



【図17】

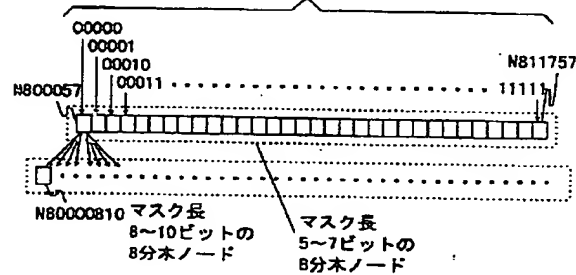
図17



【図20】

図20

宛先アドレス第0~4ビットの値により選択



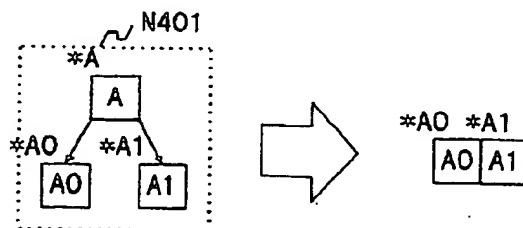
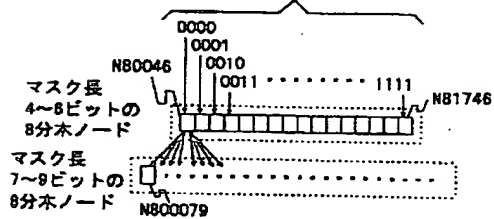
【図22】

図22

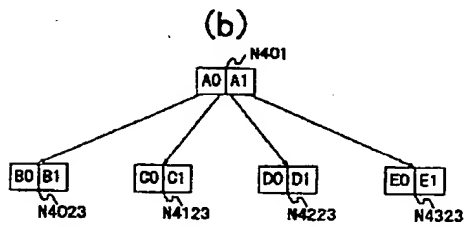
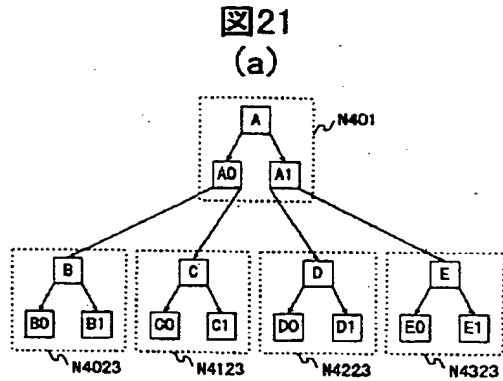
【図19】

図19

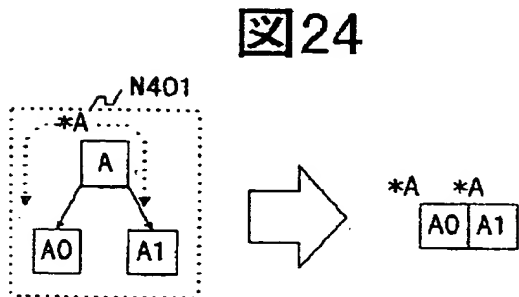
宛先アドレス第0~3ビットの値により選択



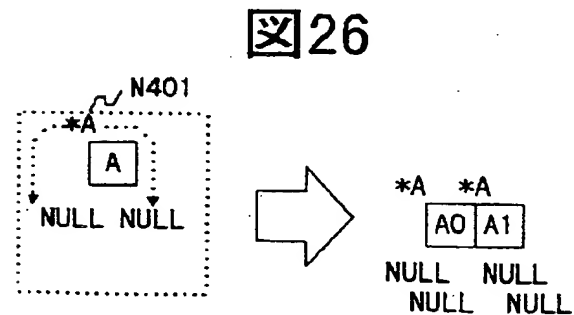
【図21】



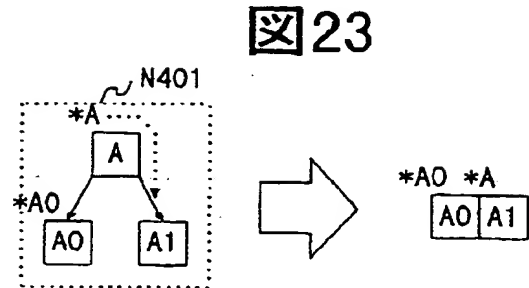
【図24】



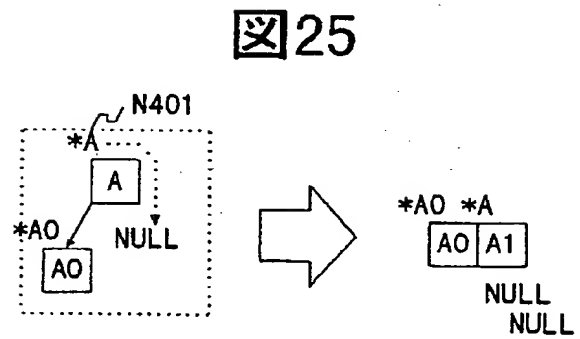
【図26】



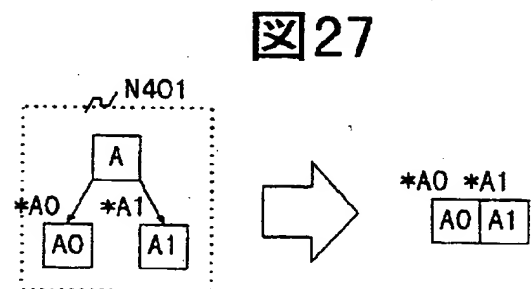
【図23】



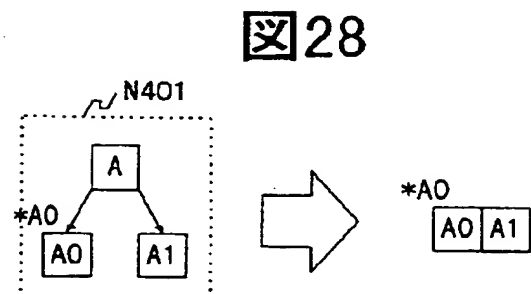
【図25】



【図27】

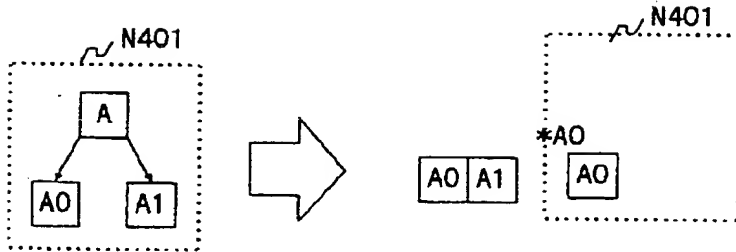


【図28】



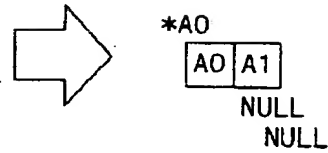
【図29】

図29



【図30】

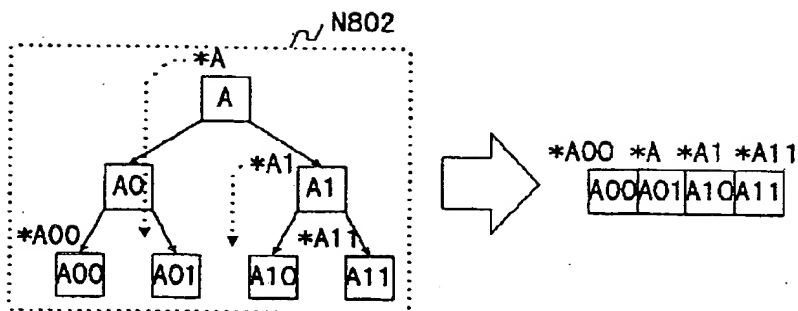
図30



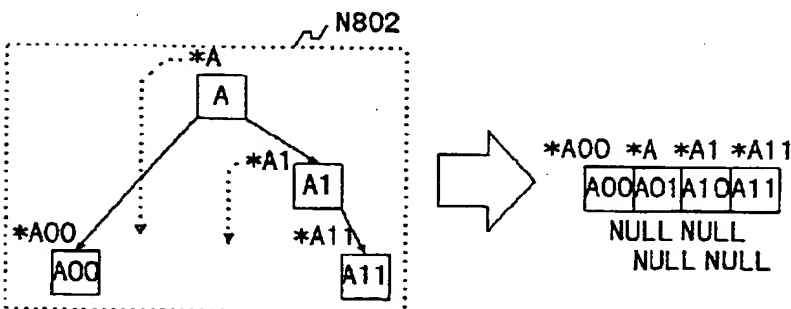
【図31】

図31

(a)

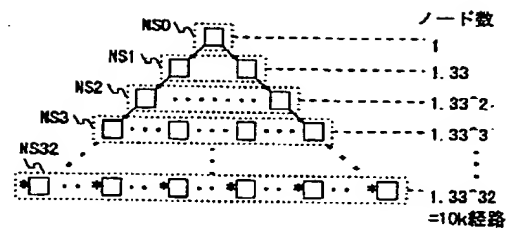


(b)



【図32】

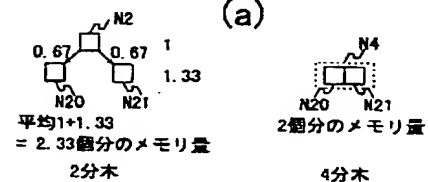
図32



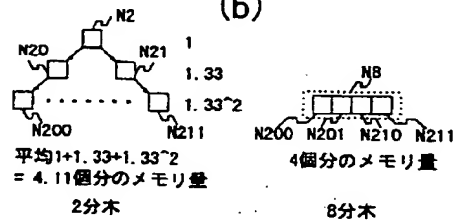
【図33】

図33

(a)



(b)



【図34】

図34

32ビット

W0	次のノードの マスク長 0	Flag 0	次のノードへのポインタ 0
W1	次のノードの マスク長 1	Flag 1	次のノードへのポインタ 1
W2	サブネットワークアドレス		
W3	次ホップアドレス		
W4	出力ポート番号		

2分本ノード

【図35】

図35

32ビット

W0	次のノードの マスク長 00	Flag 00	次のノードへのポインタ 00
W1	次のノードの マスク長 01	Flag 01	次のノードへのポインタ 01
W2	次のノードの マスク長 10	Flag 10	次のノードへのポインタ 10
W3	次のノードの マスク長 11	Flag 11	次のノードへのポインタ 11
W4	サブネットワークアドレス		
W5	出力ポート番号0		出力ポート番号1
W6	次ホップアドレス0		
W7	次ホップアドレス1		

4分本ノード

【図36】

図36

検査ビット

				00	01	10	11
W0	次のノードの マスク長 00	Flag 00	次のノードへのポインタ 00	0			
W1	次のノードの マスク長 01	Flag 01	次のノードへのポインタ 01		1		
W2	次のノードの マスク長 10	Flag 10	次のノードへのポインタ 10			2	
W3	次のノードの マスク長 11	Flag 11	次のノードへのポインタ 11				3
W4	サブネットワークアドレス			4	4	4	4
W5	出力ポート番号0		出力ポート番号1	5	5	5	5
W6	次ホップアドレス0			6	6		
W7	次ホップアドレス1					7	7

4分本ノード

【図37】

図37

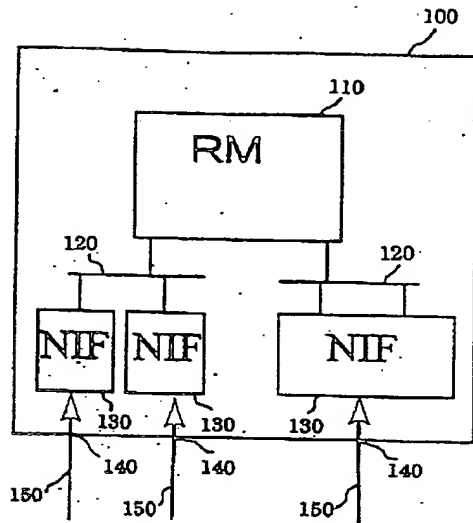
検査ビット

				00	01	10	11
W0	次のノードの マスク長 00	Flag 00	次のノードへのポインタ 00	0			
W1	次のノードの マスク長 01	Flag 01	次のノードへのポインタ 01		1		
W2	次のノードの マスク長 10	Flag 10	次のノードへのポインタ 10			2	
W3	次のノードの マスク長 11	Flag 11	次のノードへのポインタ 11				3
W4	サブネットワークアドレス			4	4	4	4
W5	出力ポート番号0		出力ポート番号1				
W6	次ホップアドレス0						
W7	次ホップアドレス1						

4分本ノード

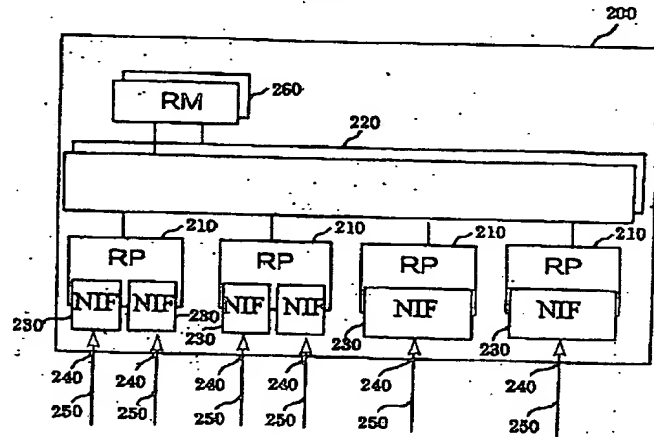
【図38】

図38



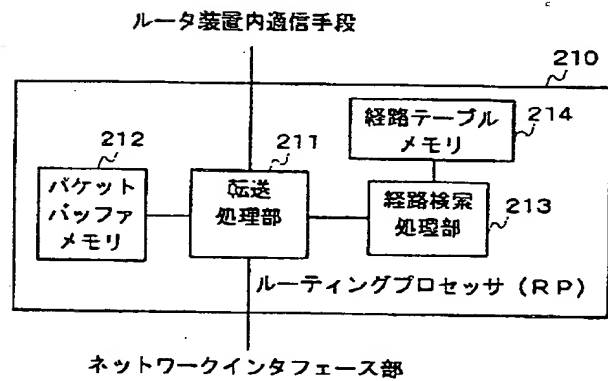
【図39】

図39

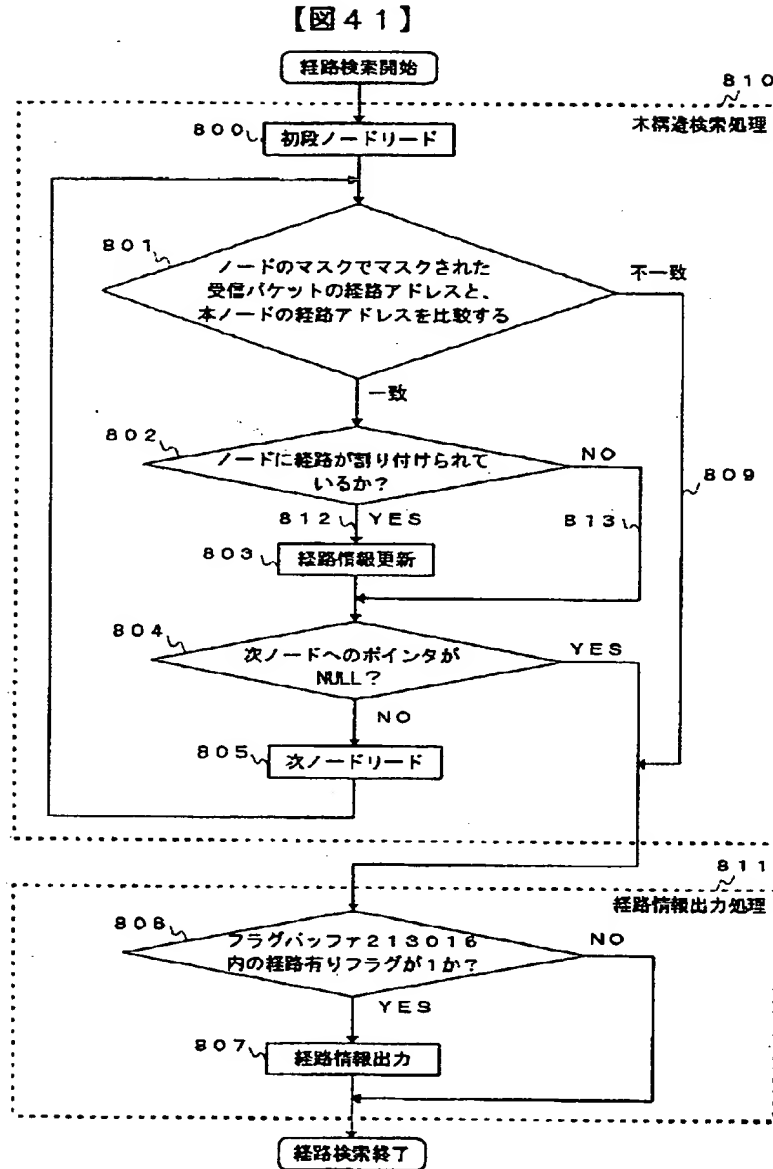


【図40】

【図40】

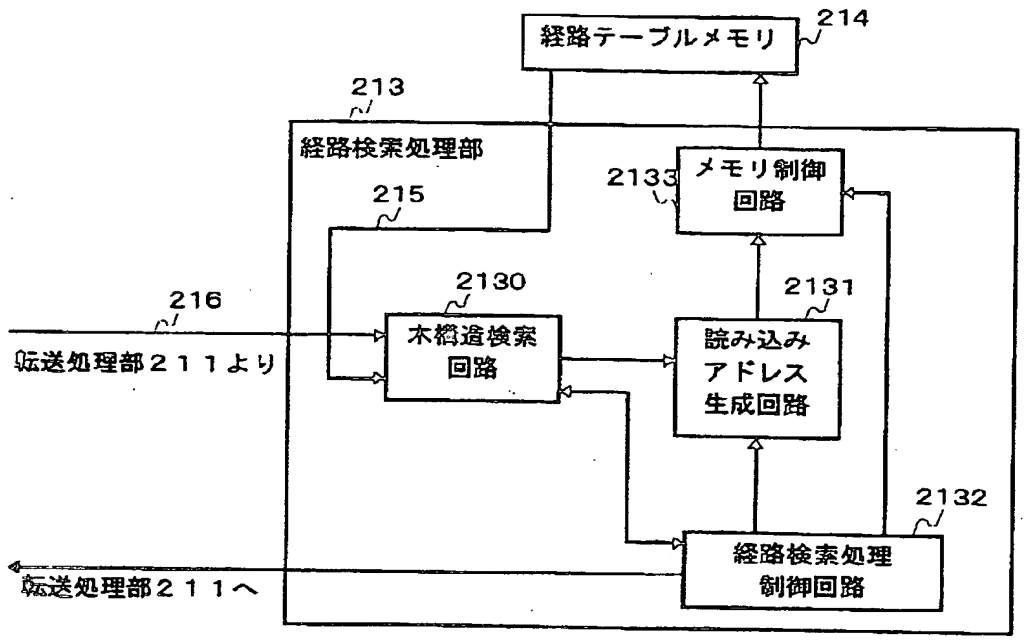


【図41】



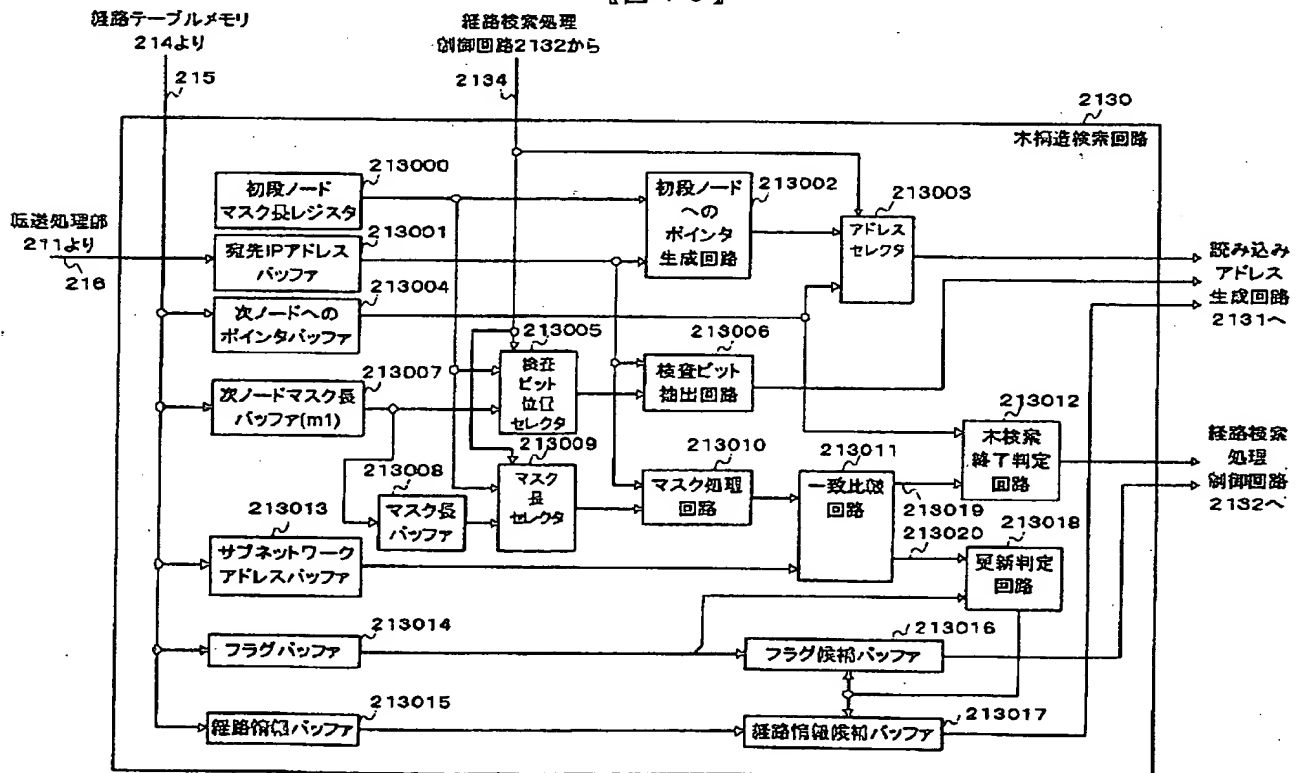
【図42】

【図42】



【図43】

【図43】



フロントページの続き

(72)発明者 赤羽 真一
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 左古 義人
神奈川県秦野市堀山下1番地 株式会社日
立製作所汎用コンピュータ事業部内
(72)発明者 田那邊 昇
神奈川県秦野市堀山下1番地 株式会社日
立製作所汎用コンピュータ事業部内